Proceedings of the 8th
World Congress on Intelligent Control and Automation
July 6-9 2010, Jinan, China

Design and Nonlinear Control of an Indoor Quadrotor Flying Robot*

Zheng Fang[1,3]
fangzheng@mail.neu.edu.cn
[1]*Key Laboratory of Integrated Automation of Process Industry, Ministry of Education*
Northeastern University
Shenyang, Liaoning Province, China

Xiu-ying Wang[2]
bywxy@126.com
[2]*College of Information Science and Technology*
Qingdao University of Science and Technology
Qingdao, Shangdong Province, China

Jian Sun[4]
sun.j@neusoft.com
[3]*National Metallurgical Automation Engineering Technology Research Center (Shenyang)*
Northeastern University
Shenyang, Liaoning Province, China
[4]*Neusoft Positron Medical System Co.,Ltd*
Neusoft Corporation
Shenyang,Liaoning Province, China

Abstract **- Micro Vertical Take-off and Landing (VTOL) systems represent a useful class of flying robots because of their strong capabilities for small-area monitoring and building exploration. In this paper, we describe a small quadrotor flying robot for the use of indoor autonomous navigation and exploration. Three-level hardware architecture is developed and a fusion of different sensors is implemented, which enables automatic hovering, control from Remote-Controller, localization and navigation in indoor environment. Since the flying robot is a nonlinear underactuated system, an Integral-Backstepping control algorithm is proposed to realize robust attitude and position control. Both simulation and experiment results are presented to validate the performance of the proposed control algorithm.**

Index Terms – flying robot, quadrotor, nonlinear control, Integral- Backstepping.

I. INTRODUCTION

In the past two decades, many autonomous ground vehicles have been developed since their wide practical application prospects, such as rescue, exploration, surveillance etc, in outdoor or indoor environment. In fact, in spite of the dramatic progress in the development of autonomous ground vehicles in recent years, navigation in unstructured natural environment still poses significant challenges. The existing ground vehicles have inherent limitations to reach the desired locations in many applications. Thus, in many cases, the use of unmanned aerial vehicles (UAV) is the only effective way to reach target to get information or to deploy instrumentation. In the last ten years, UAVs have improved their autonomy dramatically with the development of GPS position technology, inertial navigation technology, communication and control technology, and image processing technology. Thus, today we can consider some UAVs as intelligent robotic systems integrating perception, learning, real-time control, reasoning, decision making and planning capabilities for operating in complex environments.

Today, many UAVs have been developed all around the world. Most of them, however, are not suitable for indoor usage since their large body size, big noise and GPS based position technology (GPS signal is very bad in indoor environments). In recent years, with the development of new energy technology, Micro Electro Mechanical Systems (MEMS), Miniature Inertial Measurement Unit (MIMU), small quadrotor has been developing quickly and has attracted many interests in the research community [1][2][6][9]. A quadrotor aircraft is an UAV which has VTOL characteristic like helicopters but lifted and propelled by four parallel bladed rotors. Comparing with other UAVs, quadrotor has some advantages, such as high payload/volume ratio, good hovering ability, low noise and easy maneuverability. The idea of constructing a quadrotor helicopter is not new; the first attempt can date back to 1907 when the Gyroplane No. 1 of Louis and Jacques (France) was built [8]. But due to the inefficiency of power, sensors and control technologies, the quadrotor has not been actually put in practice over the last fifty years. However, quadrotor arrangements possess some advantages over the conventional ones that can be utilized in the field of small-form UAVs: by using four rotors that can rotate with individually controllable speeds make unnecessary the alteration of the rotor blade's incidence angle, hence the mechanical structure of the helicopter becomes quite simple. Therefore, it is an ideal flying robot platform for indoor applications. Besides, its simple structure, unique flight mode (like UFO), low-cost, wide application prospects and big potential commercial value make it become a hot research topic among robotic, control and aviation community in recent years.

Recently, numerous robot and control laboratories initiated projects with the purpose to construct small-form quadrotor robot and to solve the control problems raised in association with them [1] [5-8]. Like classical helicopters, quadrotor has a non-linear dynamics. Thus, in order to fully exploit its agility and to achieve a wide envelope of allowed

* Supported by National Basic Research Program of China (973Program) (2009CB320601), National 863 Program (2007BAE25B01), National R&D Project (2007AA041404, 2007AA041405), National Natural Science Foundation of China (60821063, 60828007), and the 111 Project (B08015).

978-1-4244-6712-9/10/$26.00 ©2010 IEEE

maneuvers, a control system capable of dealing with such non-linearity, measurement disturbances and noises, sensor biases and un-modeled dynamics of actuators is needed.

In this paper, we want to design a small quadrotor flying robot intended for the research of autonomous navigation, exploration and mapping in GPS-denied indoor environments. In the following chapters, it begins with the description of the construction of the quadrotor flying robot, and then the dynamic model of the quadrotor is established. After that, an Integral-Backstepping control scheme is designed, and finally some simulation and experiment results are given.

II. System Design

The quadrotor flying robot is a four-rotor aerial platform that is capable of VTOL. The concept of the quadrotor flying robot is illustrated in Fig. 1. The actuator system has two pairs of counter-rotating, fixed-pitch blades located at the four corners of the vehicle [8]. The attitude and the movement of the quadrotor can be controlled by properly changing the revolution of the rotors that results in different thrust and torsion as shown in Fig. 2. The rotors are driven by brushless motors and their speed can efficiently be controlled by local brushless motor controllers.

Since our quadrotor flying robot is intended for indoor usage, therefore following issues should be carefully considered when designing the flying robot [6].

1) The size of the aerial robot should be small enough to manoeuvre in an indoor environment; but it should be big enough to carry the high complicated on-board electronics for realizing stable flight control.
2) The real-time computation speed of the MCU should be fast enough for sensor fusion, state estimation and real-time intelligent control.
3) The flying robot is able to communicate with a ground station, at least in the respect of accepting commands and reporting data.
4) The flying robot should be equipped with light-weight Laser range finder or vision system for exploration and mapping.

According to the above requirements and characteristics of quadrotor, three-level control architecture is designed for our robot, namely motor control, flight control and navigation control respectively, as shown in Fig. 3.

As to motor control, four brushless motor controllers are used to control the speed of four sensorless Brushless DC motors (BLDC) which drive the propellers. The brushless controllers are running on Atmel ATMEGA8 micro-controller. Revolution or torque control is realized by using PID control technology. The four brushless motor control boards communicate with the flight control board via I^2C bus. In order to make the platform to achieve a high real-time stabilization performance, the flying robot is equipped with additional sensors and computational units on flight control board. Three MEMS Gyroscopes, three MEMS Accelerometers are used for the attitude control of the flying robot. Sensors data fusion as well as attitude control is done by means of an on-board 20MHz microcontroller (Atmel ATMEGA644p). In order to make the flying robot suitable for indoor scenarios, an IR distance Sensor and a light-weight Micro Laser ranger finder are connected to the navigation board. These sensors will help the flying robot to determine its position and altitude in indoor environments. Rotation along central point (yaw) is eliminated with adding a 3D digital compass, which is an important requirement for navigation. It allows us to fix the robot's orientation in space. The quadrotor can be controlled via a 72 MHz digital radio link from digital remote control unit.

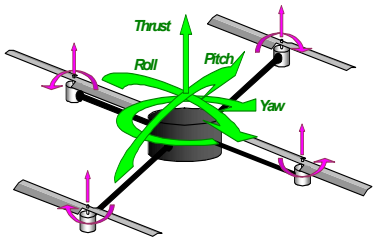


Fig. 1 Quadrotor Concept

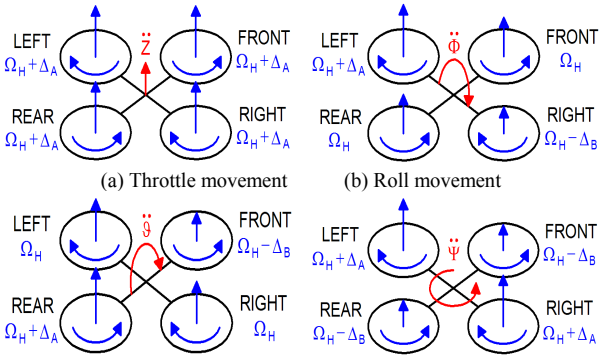


Fig. 2 Flight mode of quadrotor

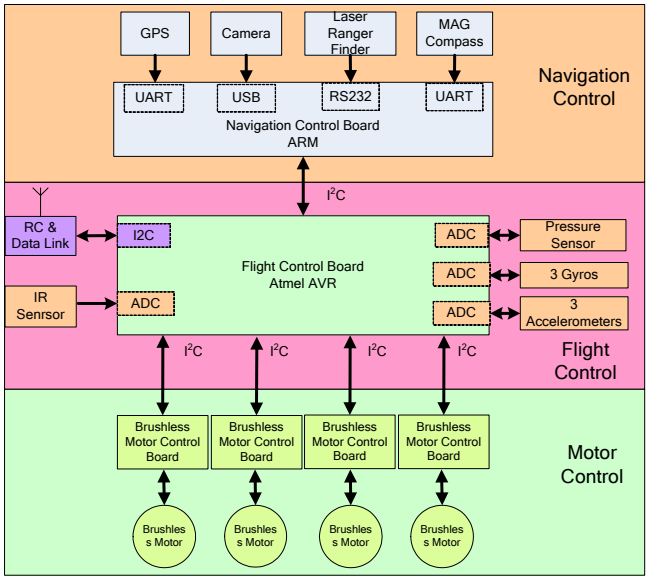


Fig. 3 Hardware architecture for real-time application

Currently, we have developed a prototype quadrotor flying robot. The flying robot has a size of $450\times450\times200$ mm ($L\times W\times H$) and a weight of about 550g (excluding Micro Laser ranger finder). With fully loaded batteries (2200mAh) it can operate for approx. 15~20min. Its maximum payload is about 300g; therefore it is strong enough for the flying robot to add additional sensors such as digital camera (about 120g) and HOKUYO Laser ranger finder (about 160g). The prototype of our quadrotor flying robot is shown in Fig. 4.

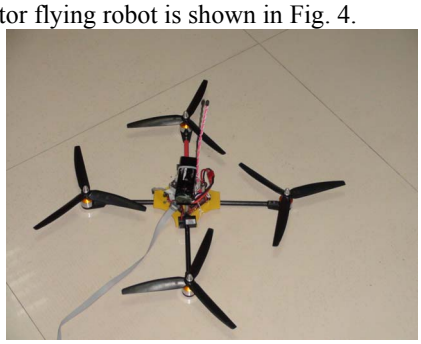


Fig. 4 Prototype of quadrotor flying robot

Since the quadrotor has 6 DOF but only four control inputs, therefore it is an under-actuated system. Besides, its high nonlinearity, together with coupled dynamics makes it an interesting platform for control researchers. In the next section, we will describe the dynamic model of the quadrotor and design an efficient control algorithm for real-time control.

III. System Dynamic Modeling

Let us consider earth fixed frame E and body fixed frame B, as shown in Fig. 5.

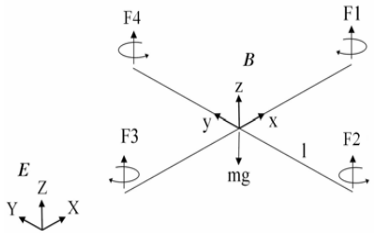


Fig. 5 Reference frames

The dynamics of any rigid body which is subject to external forces and moments can be written as:

$$\begin{aligned} \sum F_b &= m\dot{v} - mv\times\omega \\ \sum M &= I\dot{\omega} \end{aligned} \tag{1}$$

where v is body linear velocity, expressed in the body frame, $\omega=[p,q,r]$ is body angular velocity, m and I are the body mass and the moment of inertia matrix, and F_b and M are the forces and moment acting on the body. Given the position vector $\xi=[x,y,z]$ and the attitude vector $\eta=[\phi,\theta,\psi]$ composed of the three Euler angles roll, pitch and yaw, position and attitude dynamics is:

$$\begin{aligned} \dot{\xi} &= R\cdot v \\ \dot{\eta} &= T^{-1}\cdot\omega \end{aligned} \tag{2}$$

where, R and T are the rotation matrix and transformation matrix respectively:

$$R=\begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}$$

$$T=\begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & c\theta s\phi \\ 0 & -s\phi & c\theta c\phi \end{bmatrix}$$

where, c and s are abbreviation of cos and sin function.

The four propellers, rotating at angular velocity Ω_i, produce four forces F_i directed upward and applied at the propeller hub and, at the same time, four counteracting torques of strength βF_i which are directed opposite to the propeller rotating direction. Given L_i the lever arm of the four forces F_i and $\Omega_r=\Omega_2+\Omega_4-\Omega_1-\Omega_3$ the signed sum of the angular velocities of the propellers, equation (1) can be rewritten as:

$$\begin{aligned} &\sum_i (F_i\times L_i) + \beta(-F_1+F_2-F_3+F_4) - I\omega\times\omega - I_r\Omega_r\times\omega = I\dot{\omega} \\ &-mR^T g + \sum_i F_i = m\dot{v} - mv\times\omega \end{aligned} \tag{3}$$

where $I_r\Omega_r$ is gyroscopic forces, β is thrust/weight ratio and g is the gravity acceleration vector [2][8].

During hovering or navigation with moderate velocity, the roll and pitch angles remain adequately near zero degrees to allow to approximate $\sin\phi\approx\phi$ $\sin\theta\approx\theta$ and $\cos\phi\approx1$ $\cos\theta\approx1$ as well. Under these approximations, matrix T becomes an identity matrix, thus vector of derivatives of the Euler angles $\dot{\eta}$ can be approximated by the body axis angular velocity ω, yields:

$$\begin{aligned} m\ddot{\xi} &= RF_b \\ T\ddot{\eta} &= I\dot{\omega} \end{aligned} \tag{4}$$

So, we can get the following results:

$$\begin{aligned} \ddot{\xi} &= -g + R/m(\sum_i F_i) \\ I\ddot{\eta} &= -\omega\times I\omega - I_r\Omega_r\times\omega + \sum_i (F_i\times L_i) + \beta(F_2+F_4-F_1-F_3) \end{aligned} \tag{5}$$

The use of the four forces F_i as input to the system is somewhat counterintuitive. Thus, it is common to use a control allocation scheme that transforms the forces F_i into a vertical thrust U_1 and three virtual toques around the three orthogonal body axes U_2, U_3 and U_4 for Roll, Pitch and Yaw moment respectively:

$$\begin{aligned} U_1 &= F_1+F_2+F_3+F_4 = b(\Omega_1^2+\Omega_2^2+\Omega_3^2+\Omega_4^2) \\ U_2 &= F_4-F_2 = b(\Omega_4^2+\Omega_2^2) \\ U_3 &= F_3-F_1 = b(\Omega_3^2+\Omega_1^2) \\ U_4 &= F_2+F_4-F_1-F_3 = \beta b(\Omega_2^2+\Omega_4^2-\Omega_1^2-\Omega_3^2) \end{aligned} \tag{6}$$

Then, equation (5) can be rewritten and expanded into:

$$
\begin{aligned}
\ddot{x} &= (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)\frac{U_1}{m} \\
\ddot{y} &= (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)\frac{U_1}{m} \\
\ddot{z} &= -g + (\cos\phi\cos\theta)\frac{U_1}{m} \\
\ddot{\phi} &= \dot{\theta}\dot{\psi}(\frac{I_y - I_z}{I_x}) - \frac{I_r}{I_x}\dot{\theta}\Omega_r + \frac{l}{I_x}U_1 \\
\ddot{\theta} &= \dot{\phi}\dot{\psi}(\frac{I_z - I_x}{I_y}) + \frac{I_r}{I_y}\dot{\phi}\Omega_r + \frac{l}{I_y}U_2 \\
\ddot{\psi} &= \dot{\phi}\dot{\theta}(\frac{I_x - I_y}{I_z}) + \frac{1}{I_z}U_3
\end{aligned}
\tag{7}
$$

where I_x, I_y, I_z are body moment of inertia around the x, y, z axis respectively.

IV. Controller Design

For clarity of notation, the dynamics of quadrotor in (7) is described in standard state space equations $\dot{x} = f(x,u)$.

Given the state vector:

$$x = [\phi\ \dot{\phi}\ \theta\ \dot{\theta}\ \psi\ \dot{\psi}\ z\ \dot{z}\ x\ \dot{x}\ y\ \dot{y}]$$

And the input vector:

$$u = [U_1\ U_2\ U_3\ U_4]$$

The state space equations can be written as:

$$
f(x,u) = \begin{bmatrix}
\dot{\phi} \\
\dot{\theta}\dot{\psi}a_1 + \dot{\theta}a_2\Omega_r + b_1U_2 \\
\dot{\theta} \\
\dot{\phi}\dot{\psi}a_3 + \dot{\phi}a_4\Omega + b_2U_3 \\
\dot{\psi} \\
\dot{\theta}\dot{\phi}a_5 + b_3U_4 \\
\dot{z} \\
-g + (\cos\phi\cos\theta)\frac{U_1}{m} \\
\dot{x} \\
u_x\frac{U_1}{m} \\
\dot{y} \\
u_y\frac{U_1}{m}
\end{bmatrix}
\tag{8}
$$

$$
a_1 = \frac{I_y - I_z}{I_x}\quad a_2 = \frac{-J}{I_x}\quad a_3 = \frac{I_z - I_x}{I_y}\quad a_4 = \frac{-J}{I_y}
$$

where： $a_5 = \dfrac{I_x - I_y}{I_z}\quad b_1 = \dfrac{l}{I_x}\quad b_2 = \dfrac{l}{I_y}\quad b_3 = \dfrac{1}{I_z}$

$$
\begin{aligned}
u_x &= (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) \\
u_y &= (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)
\end{aligned}
$$

From (8), we can see that the rotational motions do not depend on translational motion while the opposite is not true. Thus, double-loop control architecture is designed for the flying robot's attitude and position control, as shown in Fig. 6. The inner control loop was designed for stability and tracking of desired Euler angles, with an outer control loop for regulating the robot position. Since the state space equation of quadrotor flying robot is a strict-feedback system [3]. Therefore, an Integral-Backstepping control algorithm is adopted to realize the attitude control [4].

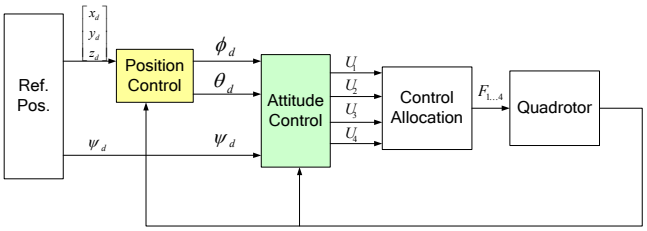


Fig. 6 Double-loop control architecture

A. Attitude Control

Three separate controllers are designed to track the desired roll, pitch, yaw angles. In the following, we will take roll control system as an example to derive the Integral-Backstepping control law.

Consider the tracking error: $e_1 = \phi_d - \phi$ and its dynamics：

$$\frac{de_1}{dt} = \dot{\phi}_d - \omega_x \tag{9}$$

The angular speed ω_x is not our control input and has its own dynamics. Therefore, we introduce a virtual control:

$$\omega_{xd} = c_1e_1 + \dot{\phi}_d + \lambda_1\chi_1 \tag{10}$$

where c_1 and λ_1 are positive constants and $\chi_1 = \int_0^t e_1(\tau)d_\tau$ is the integral of roll tracking error. Define the angular velocity tracking error as:

$$e_2 = \omega_{xd} - \omega_x \tag{11}$$

yields：

$$\frac{de_2}{dt} = c_1(\dot{\phi}_d - \omega_x) + \ddot{\phi}_d + \lambda_1e_1 - \ddot{\phi} \tag{12}$$

Using (9), (10) and (11), we can get:

$$\frac{de_1}{dt} = -c_1e_1 - \lambda_1\chi_1 + e_2 \tag{13}$$

By replacing $\ddot{\phi}$ in equation (11) by its corresponding expression from (8), yields:

$$
\begin{aligned}
\frac{de_2}{dt} &= c_1(\dot{\phi}_d - \omega_x) + \ddot{\phi}_d + \lambda_1e_1 \\
&\quad -\dot{\theta}\dot{\psi}a_1 - \dot{\theta}a_2\Omega_r - b_1U_2
\end{aligned}
\tag{14}
$$

So, using (9) (13) (14), we can get:

$$
\begin{aligned}
\frac{de_2}{dt} &= c_1(-c_1e_1 - \lambda_1\chi_1 + e_2) + \ddot{\phi}_d + \lambda_1e_1 \\
&\quad -\dot{\theta}\dot{\psi}a_1 - \dot{\theta}a_2\Omega_r - b_1U_2
\end{aligned}
\tag{15}
$$

If one chooses the control input U_2 as：

$$U_2 = \frac{1}{b_2}[(1-c_1^2+\lambda_1)e_1 + (c_1+c_2)e_2 \\ -c_1\lambda_1\chi_1 + \ddot{\phi}_d - \dot{\theta}\dot{\psi}a_1 - \dot{\theta}a_2\Omega_r] \quad (16)$$

Then the angular tracking error is:

$$\frac{de_2}{dt} = -c_2e_2 - e_1 \quad (17)$$

with c_2 is a positive constant which determines the convergence speed of the angular velocity loop. Similarly, pitch and yaw controls are:

$$U_3 = \frac{1}{b_2}[(1-c_3^2+\lambda_2)e_3 + (c_3+c_4)e_4 - c_3\lambda_2\chi_2 + \ddot{\theta}_d - \dot{\phi}\dot{\psi}a_3 + \dot{\phi}a_4\Omega_r] \\ U_4 = \frac{1}{b_3}[(1-c_5^2+\lambda_3)e_5 + (c_5+c_6)e_6 - c_5\lambda_3\chi_3] \quad (18)$$

with $(c_3, c_4, c_5, c_6, \lambda_2, \lambda_3) > 0$,and (χ_2, χ_3) are the integral position tracking error of pitch and yaw angles respectively.

The stability of the control law can be ensured by choosing the following candidate Lyapunov function:

$$V = \frac{\lambda_1}{2}\chi_1^2 + \frac{1}{2}e_1^2 + \frac{1}{2}e_2^2 \quad (19)$$

Deriving equation (10), we can get：

$$\dot{V} = \lambda_1\chi_1e_1 + e_1\frac{de_1}{dt} + e_2\frac{de_2}{dt} \quad (20)$$

Substituting equation (13) and (17) into (20) yields:

$$\dot{V} = -c_1e_1^2 - c_2e_2^2 \le 0 \quad (21)$$

So, asymptotic stability is ensured from the positive definition of V and the fact that $\dot{V}(e_1, e_2) < 0 \ \ \forall (e_1, e_2) \neq 0$ and $\dot{V}(0) = 0$ by applying LaSalle theorem. More detailed analysis can be found in [4] [8].

B. Position Control

Similarly to attitude control, the altitude and horizontal position control are also designed using Integral-Backstepping method. We define altitude tracking error as:

$$e_7 = z_d - z \quad (22)$$

And the speed tracking error is:

$$e_8 = c_7e_7 + \dot{z}_d + \lambda_4\chi_4 - \dot{z} \quad (23)$$

Then, the control law is:

$$U_1 = \frac{m}{\cos\phi\cos\theta}[g + (1-c_7^2+\lambda_4)e_7 + (c_7+c_8)e_8 - c_7\lambda_4\chi_4] \quad (24)$$

where (c_7, c_8, λ_4) are positive constants.

Position dynamics on the horizontal plane cannot be controlled directly by using one of the inputs U_1. Horizontal movement is achieved by orienting the thrust vector towards the desired direction of motion. This is done by rolling or pitching the quadrotor in response to a deviation from y_d or x_d references respectively. Thus, the position controller outputs the attitude reference ϕ_d and θ_d, which are tracked by the attitude controller, as shown in Fig. 6. Similarly, we define position tracking errors for x and y as:

$$\begin{cases} e_9 = x_d - x \\ e_{11} = y_d - y \end{cases} \quad (25)$$

And speed tracking errors are:

$$\begin{cases} e_{10} = c_9e_9 + \dot{x}_d + \lambda_5\chi_5 - \dot{x} \\ e_{12} = c_{11}e_{11} + \dot{y}_d + \lambda_6\chi_6 - \dot{y} \end{cases} \quad (26)$$

Then, we can get the position control law:

$$\begin{cases} U_x = \frac{m}{U_1}[(1-c_9^2+\lambda_5)e_9 + (c_9+c_{10})e_{10} - c_9\lambda_5\chi_5] \\ U_y = -\frac{m}{U_1}[(1-c_{11}^2+\lambda_6)e_{11} + (c_{11}+c_{12})e_{12} - c_{11}\lambda_6\chi_6] \end{cases} \quad (27)$$

Thus, from (8), we can solve the desired roll ϕ_d and pitch θ_d angles, which are the inputs of attitude control loop.

$$\phi_d = \arcsin(U_x\sin\psi - U_y\cos\psi) \\ \theta_d = \arcsin(\frac{U_x\cos\psi + U_y\sin\psi}{\cos\phi}) \quad (28)$$

V. Experiment Results

In order to evaluate the proposed control law, several simulations and experiments have been performed.

In the simulation, the models of the flight dynamics have been implemented in Matlab/Simulink. In order to simulate the sensor bias, noise, delay of practical system, following disturbances have been added: the motor dynamics are delayed and bounded, the measured angles are overlaid with an additive white noise and the control output is bounded.

In the first simulation, control laws have to stabilize the attitude of the flying robot, bringing it from an initially pose ($\phi = 0.5rad, \theta = -0.5rad, \psi = 0$) to a horizontal pose ($\phi = \theta = \psi = 0$). Fig. 7 shows the simulation results of the roll, pitch, and yaw angles. Steady state value, which should be zero for roll and pitch during hovering, is not zero due to the sensor biases.

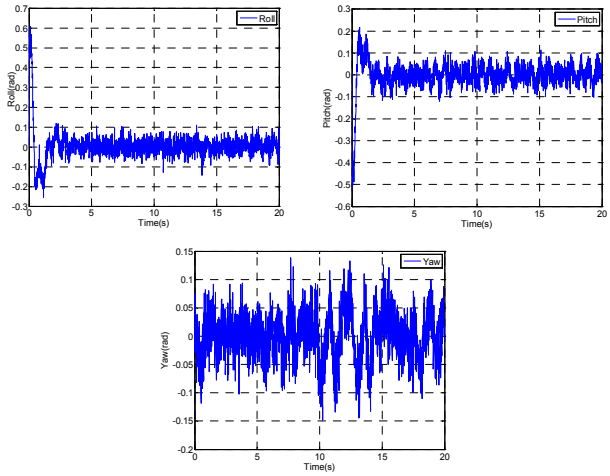
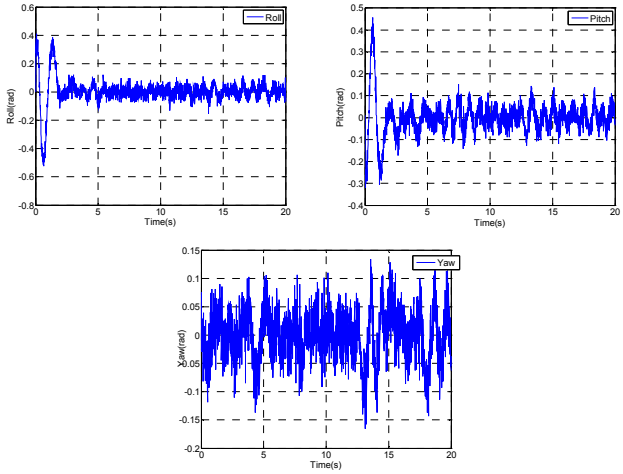


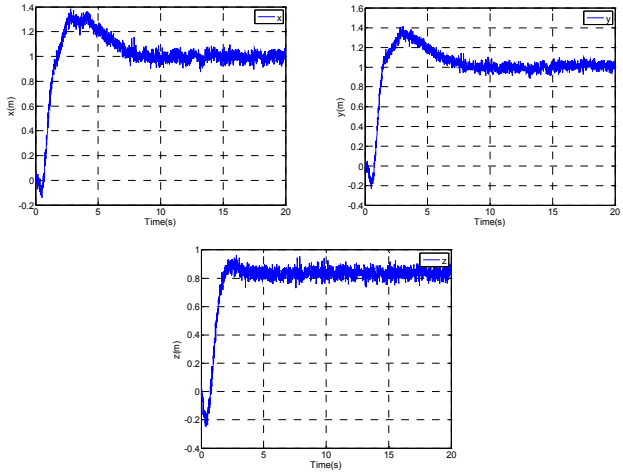
Fig. 7 Roll, pitch and yaw angles during attitude stabilization simulation. Controller parameters: $\lambda_1 = \lambda_2 = \lambda_3 = 3, c_1 = c_3 = 10, c_2 = c_4 = 5, c_5 = c_6 = 2$

The second simulation shows the integrated performance of attitude control and position control. The robot flies from an initial position to a target position. The initial position of the quadrotor is $\phi = 0.3rad, \theta = -0.3rad, \psi = 0, x = y = z = 0$, and

the target position is $\phi = \theta = \psi = 0, x = y = z = 1$. The simulation results are shown in Fig. 8.



(a) Attitude angles during the flight. Controller parameters:
$\lambda_1 = \lambda_2 = \lambda_3 = 3, c_1 = c_3 = 10, c_2 = c_4 = 5, c_5 = c_6 = 2$



(b) x, y, z position during the flight. Controller parameters:
$\lambda_4 = 2, \lambda_5 = \lambda_6 = 0.5, c_7 = 3, c_8 = 2.5, c_9 = c_{11} = 3.2, c_{10} = c_{12} = 0.5$

Fig. 8 Simulation results of the set-point tracking

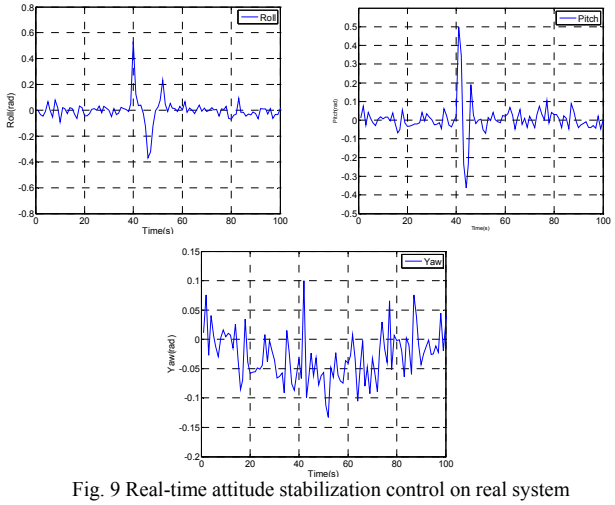


Fig. 9 Real-time attitude stabilization control on real system

The control law is also implemented on the real quadrotor flying robot. Till now, we just have completed the attitude control using Integral-Backstepping on the prototype. Fig. 9 shows some first test results obtained with the prototype, which is stabilized around $\eta = 0$. At $t = 40s$ a disturbance is added to the robot, and from the results, we can see that the robot can go back to equilibrium quickly. However, since the resolution of our gyroscopes and accelerometers are not very high, therefore the angular control accuracy still has to be improved. In the future, we will use a more accurate IMU to realize more accurate attitude and position control.

VI. Conclusion

In this paper, we have presented the hardware architecture of a quadrotor flying robot for indoor applications. The flying robot can be equipped with various sensors, such as IMU, Laser ranger finder and IR distance sensor. Therefore, it is suitable for the research of indoor autonomous navigation, exploration and mapping.

We also have developed an efficient control algorithm for the attitude stabilization and position control of the quadrotor flying robot. The performances have been analysed in terms of using various simulation and experiment results, showing the excellent behaviour of Integral-Backstepping control method. Till now, the position control experiment on the prototype haven't completed yet. In the future, we will use Laser ranger finder to test the position control performance in indoor environments. And, we will use this platform to investigate the autonomous navigation and exploration in GPS denied indoor environment.

References

1. J. Escareno, S. Salazar-Cruz and R. Lozano, “Embedded control of a four-rotor UAV,” Proceedings of the AACC American Control Conference, Minneapolis, MN, June 2006, pp.3936–3941.
2. P. Pounds, R. Mahony and P. Corke, “Modelling and Control of a Quad-Rotor Robot,” In Proceedings of the Australasian Conference on Robotics and Automation, 2006.
3. P. Adigbli, "Nolinear Attitude and Position Control of a Micro Quadrotor using Sliding Mode and Backstepping Techniques", In Proceedings of the 3rd US-European Competition and Workshop on Micro Air Vehicle Systems & EMAV 2007, September 2007, Toulouse, France.
4. Y. Tan et al., “Advanced nonlinear control strategy for motion control systems,” in Proc. of (IEEE) Power Electronics and Motion Control Conference, (PIEMC’00), Beijing, China, 2000, pp. 116-121.
5. A. Soumelidis, P. Gaspar, G. Regula, B. Lantos, "Control of an experimental mini quad-rotor UAV". Control and Automation, 2008 16th Mediterranean Conference on. 25-27 June 2008, pp.1252-1257.
6. T. Viatcheslav and S. Hartmut. "Hardware architecture of a four-rotor UAV for USAR/WSAR scenarios". Workshop Proceedings of Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS, pp.434-443, 2008.
7. S. Bouabdallah, R. Siegwart, "Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor". Proceedings of the 2005 IEEE International Conference on Robotics and Automation Barcelona, Spain, April 2005, pp. 2247-2252.
8. S. Bouabdallah, P. Murrieri, R. Siegwart, "Design and Control of an Indoor Micro Quadrotor", 2003.
9. S. Bouabdallah and R. Siegwart. "Design and Control of a Miniature Quadrotor". Advances in Unmanned Aerial Vehicles, 2008, pp: 171-210.